

# Fuzzy Control

by Dan Simon

**Most** control systems are based on proportional integral derivative (PID) technology ("Closed-Loop Control," August 2002, p. 55). Some systems, however, require a more complex control. For such systems, one option is fuzzy control.

## Fuzzy logic

Fuzzy logic imitates the logic of human thought, which is much less rigid than the calculations computers generally perform. Consider the task of driving a car. As you drive along, you notice that the stoplight ahead is red and the car in front of you is braking. Your (very rapid) thought process might be something like this: "I see that I need to stop. The road is wet because it's raining. The car is only a short distance in front of me. Therefore, I need to apply significant pressure to the brake pedal immediately." This reasoning takes place subconsciously, of course, but that's the way our brains work—in fuzzy terms.

Human brains do not base such decisions on the precise distance to the car ahead or the exact coefficient of friction between the tires and the road, as an embedded computer might. Likewise, our brains do not use a Kalman filter to derive the optimal pressure that should be applied to the brakes at a given moment. Our brains use common-sense rules, which seems to work pretty well.

But when we finally get around to pressing the brake pedal, we apply an exact force, let's say 23.26 pounds. So

although we reason in fuzzy terms, our final actions are considerably less so. The process of translating the results of fuzzy reasoning to a nonfuzzy action is called *defuzzification*.

Let's think about how a fuzzy cruise control system might work. The cruise controller maintains a constant vehicle speed in spite of neverending changes in road grade, wind resistance, and other variables. The controller does this by comparing the commanded speed with the actual speed. We can call the difference between commanded and actual speed *current error*. The *error change* is the difference in error from one sample period to the next.

If the current error is a small positive number—vehicle speed is slower than commanded—the controller needs to slightly increase the throttle angle in order to speed up the vehicle appropriately.

If both current error and error change are positive, the vehicle is going too slowly and decelerating. In this case, the controller needs to increase the throttle angle by a larger amount to achieve the desired speed.

Such specifications are called *fuzzy rules*. Table 1 shows a set of fuzzy rules that might be used for a cruise controller. The controller inputs are the variables current error and error change. The output represents a fuzzy specification of how much to change the throttle.

The terms "negative," "large negative," and so on in Table 1 are not precise numbers; they're fuzzy quantities.

The engineering challenge is to define these fuzzy quantities. What does a "positive" output change mean? What is "zero" current error? The definitions of these terms are constituted by *fuzzy membership functions*.

Fuzzy membership functions quantify the degree to which some variable belongs to some class. For example, it is overly simplistic to say that someone is either tall or not tall. A range of degrees exists between those two extremes. We might say that a male who is 6'6" or taller is 100% "tall," whereas a man who is 5'6" or shorter is 0% tall. In between 5'6" and 6'6" is a continuous curve that defines the extent to which a male belongs to the "tall" class. A six-foot tall male, for example, might be "tall" at a level of 50%. That continuous curve is a fuzzy membership function. It's up to the fuzzy logic designer to specify that function.

## Defuzzification

Suppose current error is -5 mph and error change is -1 mph/s. We might define a speed error of -5 mph as 75% "negative" and 25% "large negative." Similarly, we might define an error change of -1 mph as 50% "zero" and 50% "negative."

"Negative" errors can be represented by the membership function on the left in Figure 1. That is, the error is considered *negative* if it is between -8 mph and 0 mph. The level of membership of error increases from 0 to 1.0 as error increases from 0 mph to -4 mph. The

CUT HERE

TABLE 1 Fuzzy rules for the cruise controller

		Current Error	
		Zero	Positive
Error Change	Negative	Negative	Zero
Negative	<b>Large Negative</b>	<b>Negative</b>	<b>Zero</b>
Zero	<b>Negative</b>	<b>Zero</b>	<b>Positive</b>
Positive	<b>Zero</b>	<b>Positive</b>	<b>Large Positive</b>

FIGURE 1 Fuzzy membership functions

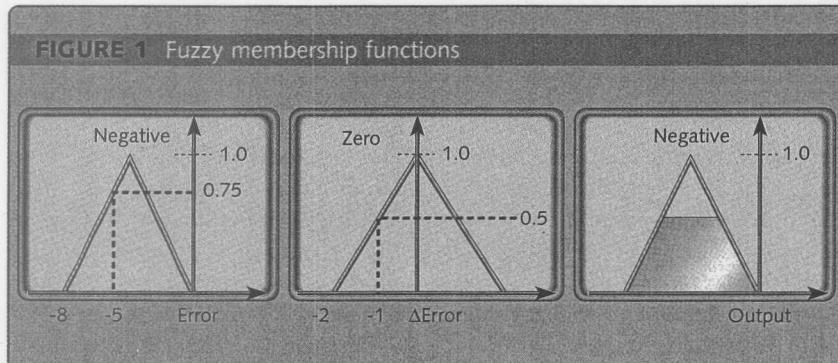
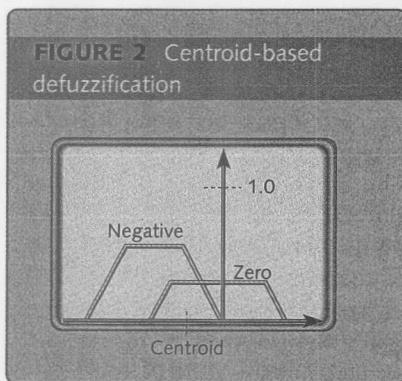


FIGURE 2 Centroid-based defuzzification



level of membership decreases from 1 to 0 as error increases from -4 mph to -5 mph; such large errors are also partial members of the "large negative" fuzzy set. An error of -5 mph belongs to the "negative" class, with a level of membership equal to 0.75.

Similarly, from the membership function in the center of Figure 1, we see that an error change of -1 mph/s belongs to the "zero" class with a level of membership equal to 0.5.

The rules in Table 1 say that if current error is "negative" and error change is "zero," then throttle change

should be "negative." The fuzzy output corresponding to "negative" might be represented by the rightmost function in Figure 1. We want a "negative" output, but our output only has a level of membership of 0.5 in the "negative" class. This is because 0.5 is the minimum contributor to this fuzzy rule. That is, since current error is 0.75 "negative" and error change is 0.5 "zero," the minimum of these two (0.5) is used as the level of membership of the output.

But that's not all. An error of -5 mph may also belong to the "zero" class with a certain level of membership. Another fuzzy rule from Table 1 says that if current error is "zero" and error change is "zero," our output should be "zero." There is a fuzzy output corresponding to "zero," similar to the fuzzy output for "negative" on the right in Figure 1. So there are actually two fuzzy outputs: a "negative" output with a 0.5 level of membership and a "zero" output with a different level of membership.

At this point we must defuzzify the output to convert our fuzzy outputs to a single number. There are many ways to defuzzify, but the most common approach is to compute the center of mass of the set of fuzzy outputs. This process is illustrated in Figure 2. The process shown there can be represented mathematically as:

$$\text{output} = \int x f(x) dx$$

where  $x$  is the value of the horizontal coordinate and  $f(x)$  is the sum of all of the fuzzy output values. Although this integral would be difficult to calculate in an embedded system, the math can be simplified considerably depending on the type of fuzzy membership functions that are used.

## When to get fuzzy

PID controllers and variations thereof work well in many control systems. But when the system to be controlled contains uncertainty or is highly complex, poorly understood, or nonlinear, fuzzy control may work better. In the example discussed here, the very nature of the problem suggests the use of fuzzy logic.

Once mastered, fuzzy logic is a valuable tool in the control engineer's repertoire.

esp

*Dan Simon is a professor in the Electrical and Computer Engineering Department at Cleveland State University. His teaching and research interests include signal processing, control theory, embedded systems, fuzzy logic, and neural networks. His e-mail address is d.j.simon@csuohio.edu.*

## Further reading

Miller, Byron. "A Top-Down Approach to Fuzzy Logic Design," *Embedded Systems Programming*, July 1998, p. 52.